Verified correctness of critical programs

Chris Hawblitzel (Operating Systems Group)
Rustan Leino (Research in Software Engineering)
Bryan Parno (Security and Privacy Research Group)

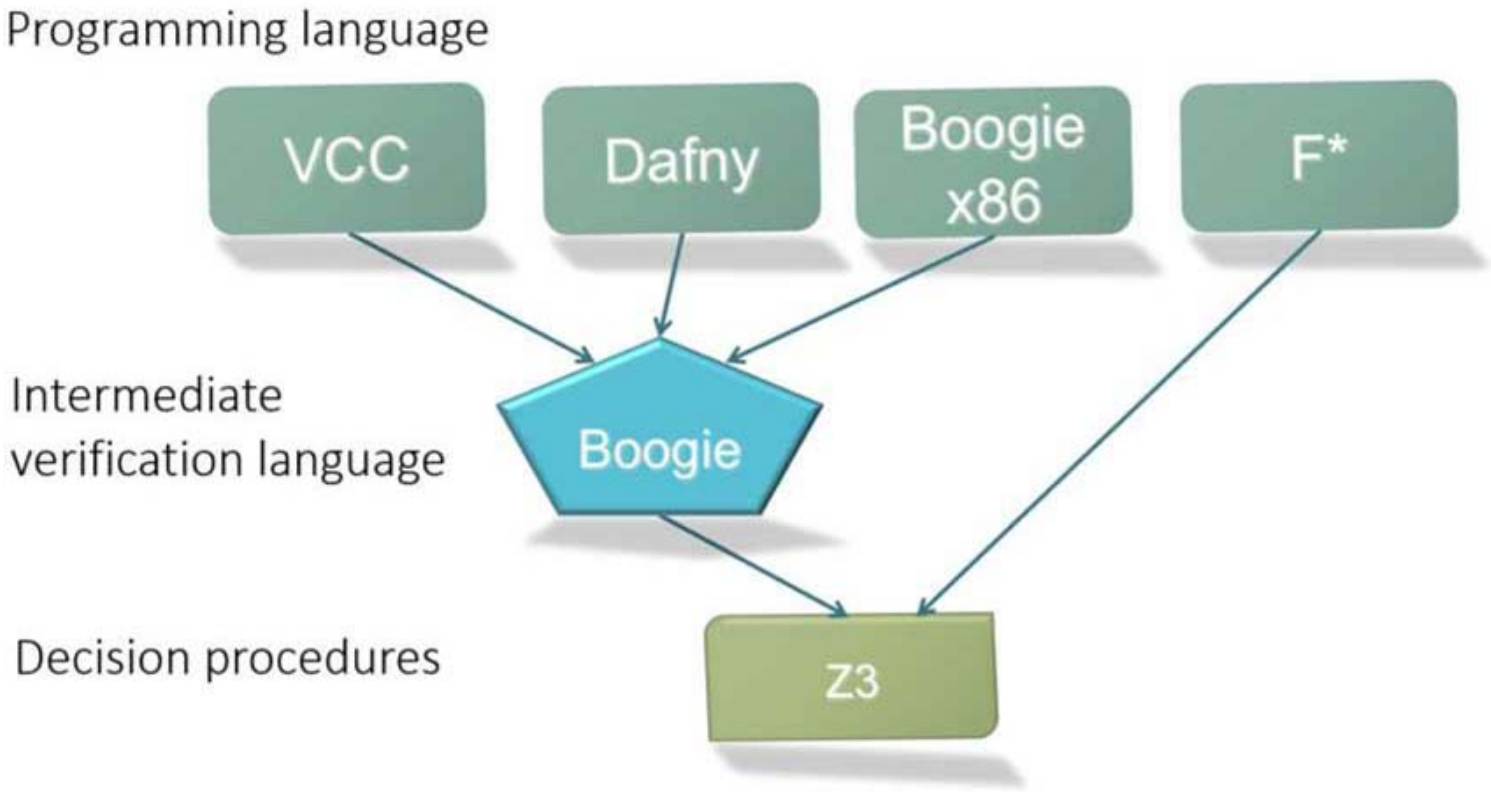
Who would verify their programs?

Target verification applications:

Research direction

*) Diagram is not to scale

Verification tools



Outline

Small demo

CloudMake verification

IronClad verification

Demo

Log

CloudMake

- Modern *make* utility
  - Functional subset of TypeScript
  - ***exec*** construct – calls out to compilers, linkers, etc.
- Algorithm correctness
  - Formalized as a program in Dafny
    - *Interpreter for language*
    - *Axioms for* ***exec***
  - Correctness properties proved
    - *Parallel builds are correct*
    - *Cache is consistent*
    - *Cached behavior = clean-build behavior*

Demo

CloudMake [joint work with Maria Christakis and Wolfram Schulte]

The Ironclad Project:
Full Verification of Security-Sensitive Services in Dafny


Chris Hawblitzel


Jon Howell


Jay Lorch


Bryan Parno

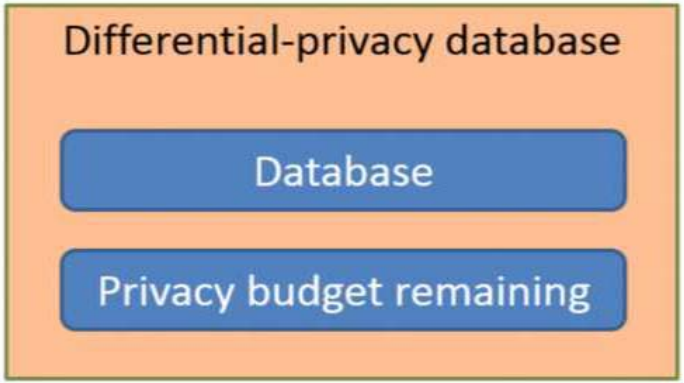

Brian Zill


Arjun Narayan


Danfeng Zhang

Two ironclad examples: append-only audit log and differential privacy

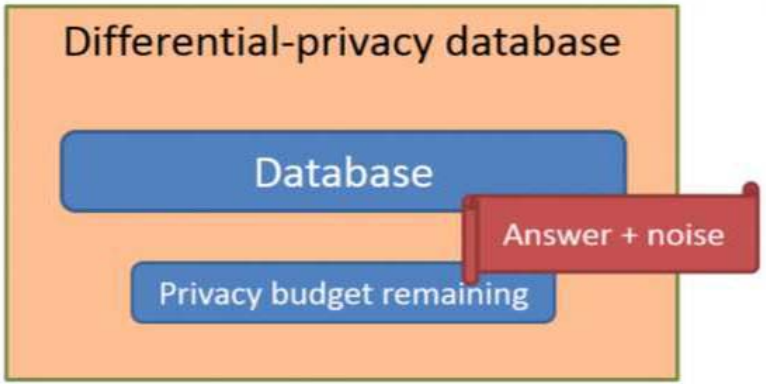
Entries cannot be removed.



Private data revealed only according to policy.

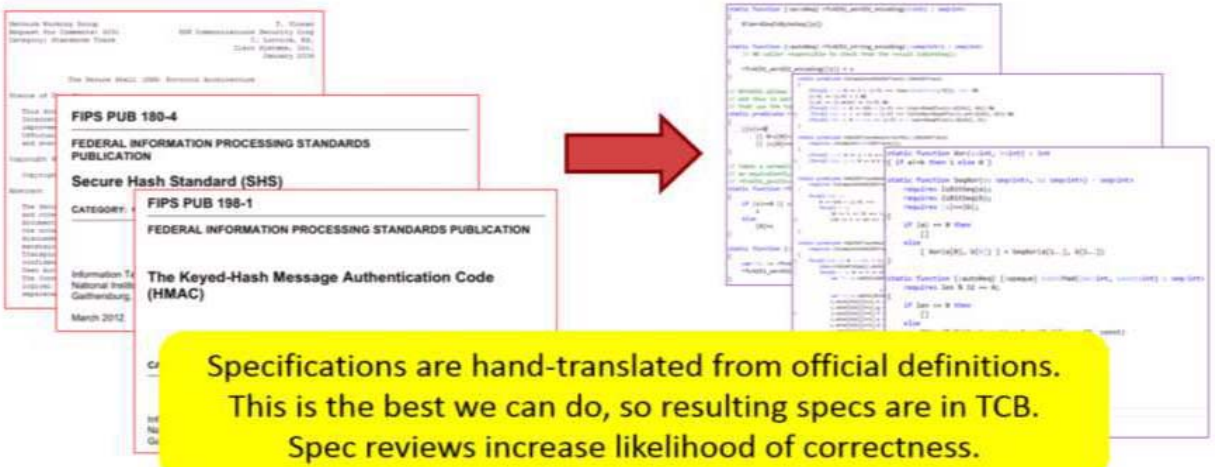
Two ironclad examples: append-only audit log and differential privacy

Entries cannot be removed.



Private data revealed only according to policy.

Security-sensitive services need cryptography, so we built a crypto library.



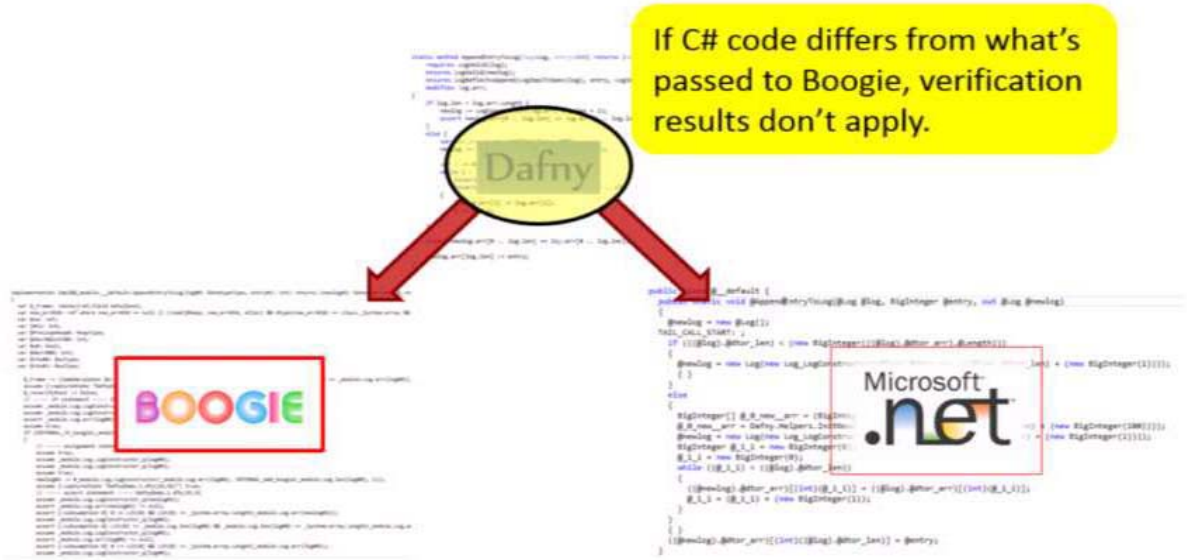
Specifications are hand-translated from official definitions.
This is the best we can do, so resulting specs are in TCB.
Spec reviews increase likelihood of correctness.

Ironclad apps written in Dafny

...but standard Dafny toolchain includes large TCB.

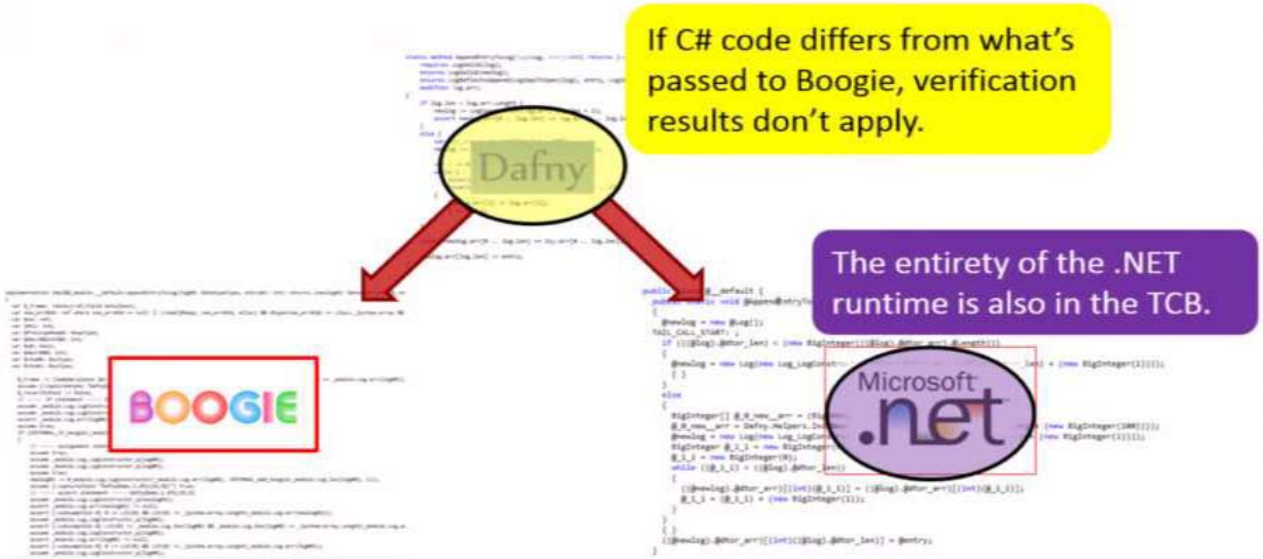
Ironclad apps written in Dafny

...but standard Dafny toolchain includes large TCB.



Ironclad apps written in Dafny

...but standard Dafny toolchain includes large TCB.



Our new Dafny compiler substantially reduces the TCB.

Our new Dafny compiler substantially reduces the TCB.

Assembly code is passed to Boogie, so verification results apply.

No dependence on .NET runtime – just x86

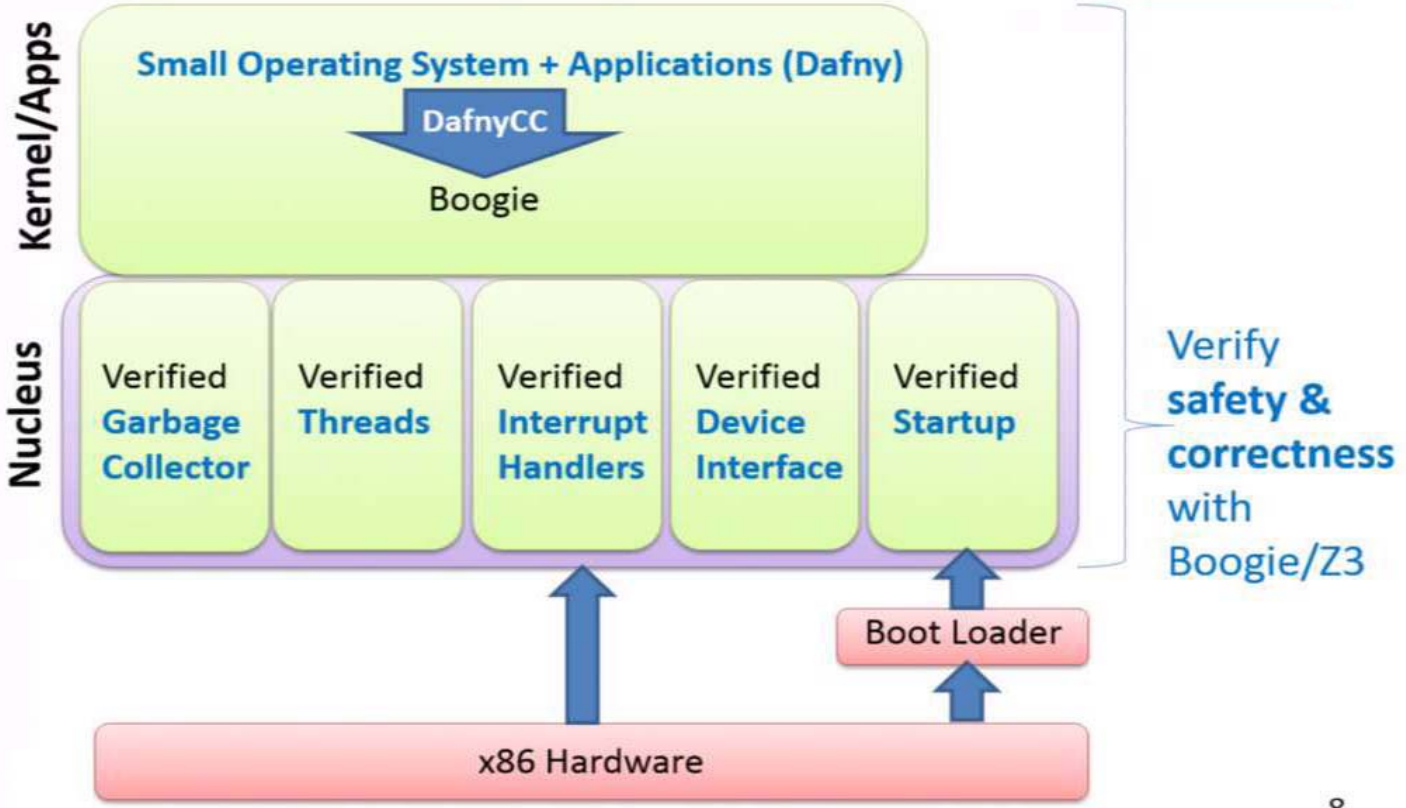
DafnyCC demo

DafnyCC features

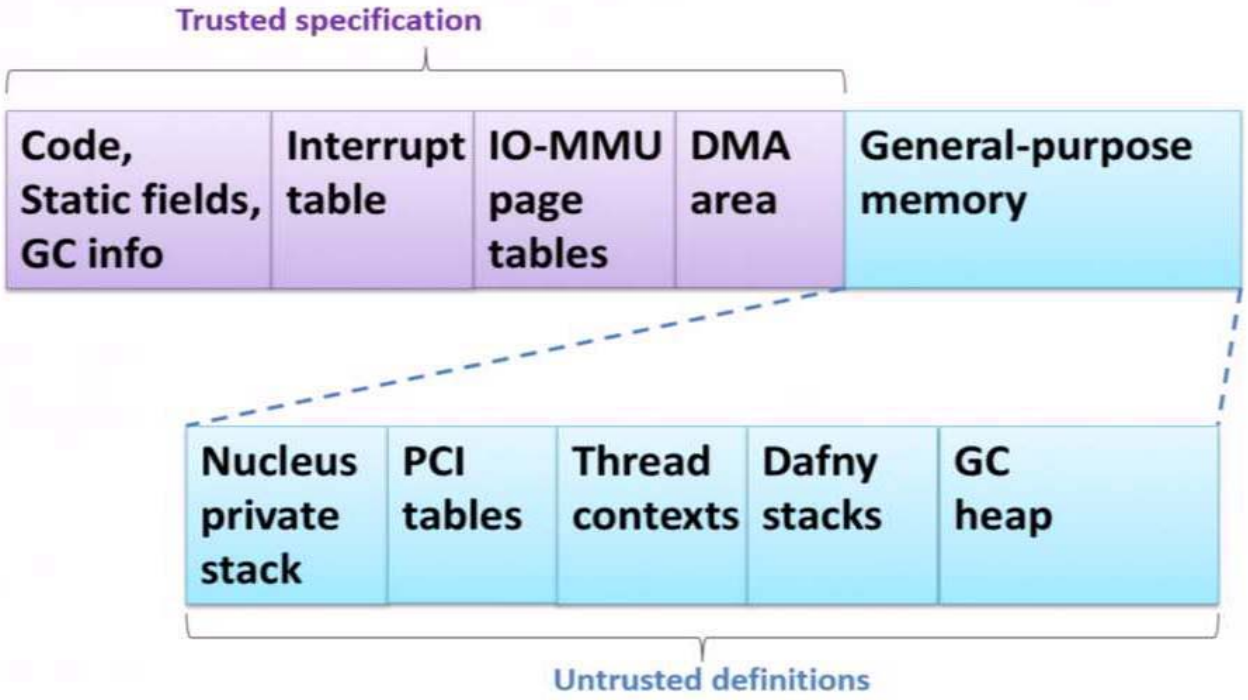
- Supports large subset of Dafny language:
  - Expressions: add, sub, bitwise, booleans, function calls, …
  - Statements: assignment, if/else, while, …
  - Types: int, bool, array-of-int, datatypes
- No object-oriented featuers
- Linear scan register allocator
- Not an optimizing compiler!

Verve: a verifiably safe OS

"every assembly language instruction checked for safety"

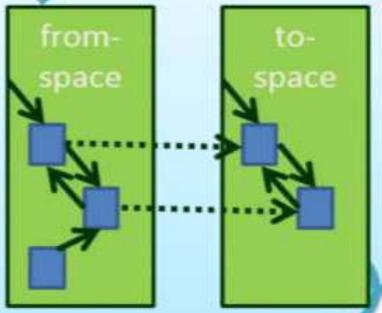


Memory management



Memory management

```
...
&& (forall i:int::{T(i)}
        T(i) && Fi <= i && i < Fk && r1[i] != NO_ABS &&
        (IsFwdPtr(gcMem[i + 4])  ==>
                Pointer(r2, gcMem[i + 4] - 4, r1[i])
            && AlignedHeapAddr(i + 4)
            && word(gcMem[i + 4]))
...
```



Nucleus private stack	PCI tables	Thread contexts	Dafny stacks	GC heap

Untrusted definitions

Ironclad progress so far

- Verve
- Libraries
  - Big integers and rationals
  - Crypto: SHA-1, SHA-256, HMAC, RSA
  - Utilities for manipulating bytes, words, sequences, arrays
  - Math
- Drivers
  - TPM
  - Network
- Services
  - Password vault
  - Notary
  - TrInc
  - Differentially-private database